AMENDMENTS TO THE CLAIMS:

Without prejudice, this listing of the claims replaces all prior versions and listings of the claims in the present application:

LISTING OF CLAIMS:

1. (Previously Presented) A method of data encryption in programming of a control unit comprising:

encrypting a complete stream of data to be transmitted in a programming unit using a first key, wherein a byte by byte encryption of the complete stream of data is capable of being performed, and wherein no byte-wise allocation between input and output data occurs;

transmitting the data that had been encrypted to the control unit via a data line; and

decrypting the data that had been encrypted in the programming unit using a second key provided in the control unit;

wherein:

successive bytes during encryption are provided with an index i, where i = 0, 1, 2, . . .,

an encrypted byte n* is formed from an unencrypted byte n according to the following, a starting value n_{-1} being used for decryption and encryption:

$$n_{-1} \equiv S_o$$

$$n_i^* = \left(n_i <<< \sum_{j=0}^{i} n_{j-1}^* \right) \oplus S_{h\left(\sum_{j=0}^{i} n_{j-1}^* \right)},$$

an unencrypted byte n is formed from an encrypted byte n* according to the following:

$$n_i = \left(n_i^* \oplus S_{h\left(\sum_{j=0}^{i} n_{j-1}^* \right)} \right) >>> \sum_{j=0}^{i} n_{j-1}^*$$

2. (Original) The method of claim 1, wherein the first key and the second key are identical.

3. (Original) The method of claim 1, wherein the first key and the second key are not identical.

4. (Original) The method of claim 2, wherein each one of the first key and the second key includes a table that is accessed by a hash function.

5. (Original) The method of claim 1, wherein at least one of the first key and the second key is implemented in an electronic circuit.

6. (Original) The method of claim 1, wherein at least one of the first key and the second key is implemented in the form of a computer program.

7. (Previously Presented) A data encryption system, comprising:

a programming unit in which a first key is provided;

a control unit in which a second key is provided; and

a data line coupled to the programming unit and the control unit for transmitting encrypted data, the encrypted data being an encryption of a complete stream of data, wherein a byte by byte encryption of the complete stream of data is capable of being performed, wherein encryption of a byte includes a rotation of bits of the byte about a number of positions, the number depending on an entire history of the encryption of the data, and wherein no byte-wise allocation between input and output data occurs;

wherein:

successive bytes during encryption are provided with an index i, where i = 0, 1, 2, . . .,

an encrypted byte n* is formed from an unencrypted byte n according to the following, a starting value n_{-1} being used for decryption and encryption:

$$n_{-1} \equiv S_o$$

$$n_i^* = \left(n_i <<< \sum_{j=0}^{i} n_{j-1}^* \right) \oplus S_{h\left(\sum_{j=0}^{i} n_{j-1}^* \right)},$$

an unencrypted byte n is formed from an encrypted byte n* according to the following:

$$n_i = \left(n_i^* \oplus S_h\left(\sum_{j=0}^{i} n_{j-1}^* \right) \right) >>> \sum_{j=0}^{i} n_{j-1}^*$$

8. (Original) The system of claim 7, wherein the first key and the second key are identical.

9. (Original) The system of claim 7, wherein the first key and the second key are not identical.

10. (Original) The system of claim 7, wherein the programming unit and the control unit each includes an electronic computing unit and a memory module that are linked together by a data bus.

11. (Currently Amended) A computer program product having program code executable by a computing unit, the program code when executed causing the computing unit to perform a method, the method comprising:

performing an encryption of a complete stream of data in accordance with a table and a hash function, wherein a byte by byte encryption of the complete stream of data is capable of being performed, and wherein no byte-wise allocation between input and output data occurs:

wherein:

successive bytes during encryption are provided with an index i, where i = 0, 1, 2, . . .,

an encrypted byte n* is formed from an unencrypted byte n according to the following, a starting value n_{-1} being used for decryption and encryption:

$$n_{-1} \equiv S_o$$

$$n_i^* = \left(n_i <<< \sum_{j=0}^{i} n_{j-1}^* \right) \oplus S_{h\left(\sum_{j=0}^{i} n_{j-1}^* \right)},$$

an unencrypted byte n is formed from an encrypted byte n* according to the following:

$$n_i = \left(n_i^* \oplus S_{h\left(\sum_{j=0}^{i} n_{j-1}^*\right)} \right) >>> \sum_{j=0}^{i} n_{j-1}^*$$

12. (Previously Presented) The computer program product of claim 11, wherein the computing unit includes an electronic computing unit in a programming unit.

13. (Canceled)

14. (Canceled).

15. (Currently Amended) A computer-readable medium, comprising:

a program code executable on a computing unit for performing an encryption of a complete stream of data in accordance with a table and a hash function, wherein a byte by byte encryption of the complete stream of data is capable of being performed, and wherein no byte-wise allocation between input and output data occurs, as provided for in the context of the claimed subject matter:

wherein:

successive bytes during encryption are provided with an index i, where i = 0, 1, 2, . . .,

an encrypted byte n* is formed from an unencrypted byte n according to the following, a starting value n_{-1} being used for decryption and encryption:

$$n_{-1} \equiv S_o$$

$$n_i^* = \left(n_i <<< \sum_{j=0}^{i} n_{j-1}^* \right) \oplus S_{h\left(\sum_{j=0}^{i} n_{j-1}^*\right)},$$

an unencrypted byte n is formed from an encrypted byte n* according to the following:

$$n_i = \left(n_i^* \oplus S_{h\left(\sum_{j=0}^{i} n_{j-1}^* \right)} \right) >>> \sum_{j=0}^{i} n_{j-1}^*$$

16. (Currently Amended) A computer-readable medium, comprising:

a program code executable on a computing unit for performing a decryption of a complete stream of data in accordance with a table and a hash function, wherein a byte by byte decryption of the complete stream of data is capable of being performed, and wherein no byte-wise allocation between input and output data occurs:

wherein:

successive bytes during encryption are provided with an index i, where i = 0, 1, 2, . . .,

an encrypted byte n* is formed from an unencrypted byte n according to the following, a starting value n_{-1} being used for decryption and encryption:

$$n_{-1} \equiv S_o$$

$$n_i^* = \left(n_i <<< \sum_{j=0}^{i} n_{j-1}^* \right) \oplus S_{h\left(\sum_{j=0}^{i} n_{j-1}^* \right)},$$

an unencrypted byte n is formed from an encrypted byte n* according to the following:

$$n_i = \left(n_i^* \oplus S_{h\left(\sum_{j=0}^{i} n_{j-1}^* \right)} \right) >>> \sum_{j=0}^{i} n_{j-1}^*$$

17. (Currently Amended) The method of claim 1, wherein there is no bit-wise allocation between input and output data:

wherein:

successive bytes during encryption are provided with an index i, where i = 0, 1, 2, . . .,

an encrypted byte n* is formed from an unencrypted byte n according to the following, a starting value n_{-1} being used for decryption and encryption:

$$n_{-1} \equiv S_o$$

$$n_i^* = \left(n_i <<< \sum_{j=0}^{i} n_{j-1}^* \right) \oplus S_{h\left(\sum_{j=0}^{i} n_{j-1}^*\right)},$$

an unencrypted byte n is formed from an encrypted byte n* according to the following:

$$n_i = \left(n_i^* \oplus S_{h\left(\sum_{j=0}^{i} n_{j-1}^*\right)} \right) >>> \sum_{j=0}^{i} n_{j-1}^*.$$

18. (Previously Presented) The method of claim 7, wherein there is no bit-wise allocation between input and output data.

19. (Previously Presented) The method of claim 11, wherein there is no bit-wise allocation between input and output data.